PCT/EP 00 / 09311

1011

Europäisches Patentamt | European Patent Office | Office européen des brevets

EP 00/09311

REC'D 13 DEC 2000
WIPO PCT

Bescheinigung Certificate Attestation

10/089021

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. Patent application No. Demande de brevet n°

99118990.3

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

PRIORITY DOCUMENT

SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

I.L.C. HATTEN-HECKMAN

DEN HAAG, DEN
THE HAGUE, 06/12/00
LA HAYE, LE

EPA/EPO/OEB Form 1014 - 02.91

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.: 99118990.3
Demande n°:

Anmeldetag:
Date of filing: 27/09/99
Date de dépôt:

Anmelder:
Applicant(s):
Demandeur(s):
DEUTSCHE THOMSON-BRANDT GMBH
78048 Villingen-Schwenningen
GERMANY

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:
Method for processing video pictures for display device

In Anspruch genommene Prioriät(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat: State: Pays:	Tag: Date: Date:	Aktenzeichen: File no. Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:
/

Am Anmeldetag benannte Vertragstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE/
Etats contractants désignés lors du depôt:

Bemerkungen:
Remarks:
Remarques:

The original title of the application reads as follows:
Method for processing video pictures for display on a display device.

Method for processing video pictures for display on a display device

The invention relates to a method for processing video pictures for display on a display device.
More specifically the invention is closely related to a kind of video processing for improving the picture quality of pictures which are displayed on matrix displays like plasma display panels (PDP) or other display devices where the pixel values control the generation of a corresponding number of small lighting pulses on the display.

The Plasma technology now makes it possible to achieve flat color panel of large size (out of the CRT limitations) and with very limited depth without any viewing angle constraints.
Referring to the last generation of european TV, a lot of work has been made to improve its picture quality. Consequently, a new technology like the Plasma one has to provide a picture quality as good or better than standard TV technology. On one hand, the Plasma technology gives the possibility of "unlimited" screen size, of attractive thickness ... but on the other hand, it generates new kinds of artefacts which could degrade the picture quality.
Most of these artefacts are different as for TV picture and that makes them more visible since people are used to seeing old TV artefacts unconsciously.
The artefact, we will study here, is called "dynamic false contour effect" since it corresponds to disturbances of gray levels and colors in the form of an apparition of colored edges in the picture when an observation point on the PDP screen moves. The degradation is most visible when the image has a smooth gradation like skin...
The next figure shows the simulation of such a false contour effect on a natural scene with skin areas :

In addition, the same problem occurs on static images when observers are moving their heads and that leads to the conclusion that such a failure depends on the human visual perception and happens on the retina...
A lot of algorithms today are based on motion estimator in order to be able to anticipate the motion of the eye to reduce or suppress this false contour effect. However, these different algorithms provide solution for false contour without any information concerning the motion estimator used.
In the past, the motion estimator evolution was mainly focused on flicker-reduction for european TV (e.g. from 50Hz to 100Hz) or proscan conversion and for compression in the scope of MPEG-encoder and so one. Nevertheless, the problems which have to be solved for such applications are different from the PDP dynamic false contour issue. For that reason, we propose in this document a way to improve the quality of the false contour reduction using standard motion detectors with a post-processing of the estimated motion-vectors according to the human visual system.

A Plasma Display Panel (PDP) utilizes a matrix array of discharge cells which could only be "ON" or "OFF". Also unlike a CRT or LCD in which gray levels are expressed by analog control of the light emission, a PDP controls the gray level by modulating the number of light pulses per frame. This time-modulation will be integrated by the eye over a period corresponding to the eye time response.

When an observation point (eye focus area) on the PDP screen moves, the eye will follow this movement. Consequently, it will no more integrate the same cell over a frame (static integration) but it will integrate information coming from different cells located on the movement trajectory and it will mix all these light pulses together which leads to a faulty signal information.

Today, a basic idea to reduce this false contour effect is to detect the movements in the picture (displacement of the eye focus area) and to apply different type of correction over this displacement in order to be sure the eye will only perceive the correct information through its movement.

Nevertheless, in the past, the motion estimator evolution was mainly focused on other applications than Plasma technology and the aim of a false contour compensation needs some adaptation above all in the use of the motion vectors coming from the motion estimator.

As already said, a dynamic false contour reduction will be applied on a movement trajectory defined by the motion vectors. Nevertheless, since the panel is a matrix array of pixels, we need to convert a motion vector to a way in this matrix with the respect of the human visual system behavior.

For that reason, we propose in this document a way to improve the quality of the false contour reduction using standard motion detectors with a post-processing of the estimated motion-vectors according to the human visual system. It could be implemented for each kind of Plasma technology at each level of its development (even if the scanning mode and sub-field distribution is not well defined).

Considering above artefacts, occurring on the retina, different inventions based on a motion estimator exist in the literature today. All these inventions will used the motion vectors coming from the motion estimator to apply different kind of compensation. Nevertheless, there is no information about the way to use these motion vectors or about a possibility to improve them.

In fact, since a PDP is a matrix of pixel, each kind of correction has to respect this matrix segmentation of the panel. Taking into account a motion vector, it will be used to determin in the matrix of pixel a trajectory to apply the compensation. For that purpose it is necessary to convert a vector in a discrete trajectory which can leads to faulty or partial compensation.

In this invention, the characteristics of the human visual system will be used to implement a post processing on the vectors coming from the motion estimator. That makes possible to define a compensation trajectory respecting the human eye in order to improve the global quality of the compensation.

This algorithm improves the efficiency of each dynamic false contour correction based on a motion estimator since it better respects the behavior of the human visual system,

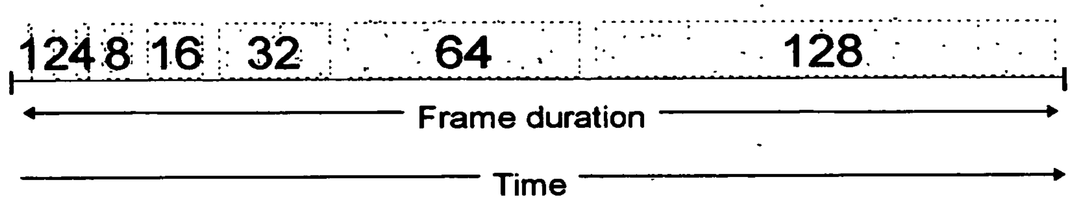
This algorithm is very simple to implement since it does not request complicated computations

As previously said, a Plasma Display Panel (PDP) utilizes a matrix array of discharge cell which can only be "ON" or "OFF". The PDP controls the gray level by modulating the number of light pulses per frame. This time modulation will be integrated by the eye over a period corresponding to the human eye time -response.

Let us assume, we want to dispose of 8 bit luminance levels, in that case each level will be represented by a combination of the 8 following bits :

1 - 2- 4 - 8 - 16 - 32 - 64 - 128

To realize such a coding with the PDP technology, the frame period will be divided in 8 lighting periods (called sub-fields), each one corresponding to a bit. The number of light pulses for the bit "2" is the double as for the bit "1"... With these 8 sub-periods, it is possible through combination, to build the 256 gray levels. Without motion, the eye of the observers will integrate over about a frame period these sub-periods and catch the impression of the right gray level. The next figure presents this decomposition :

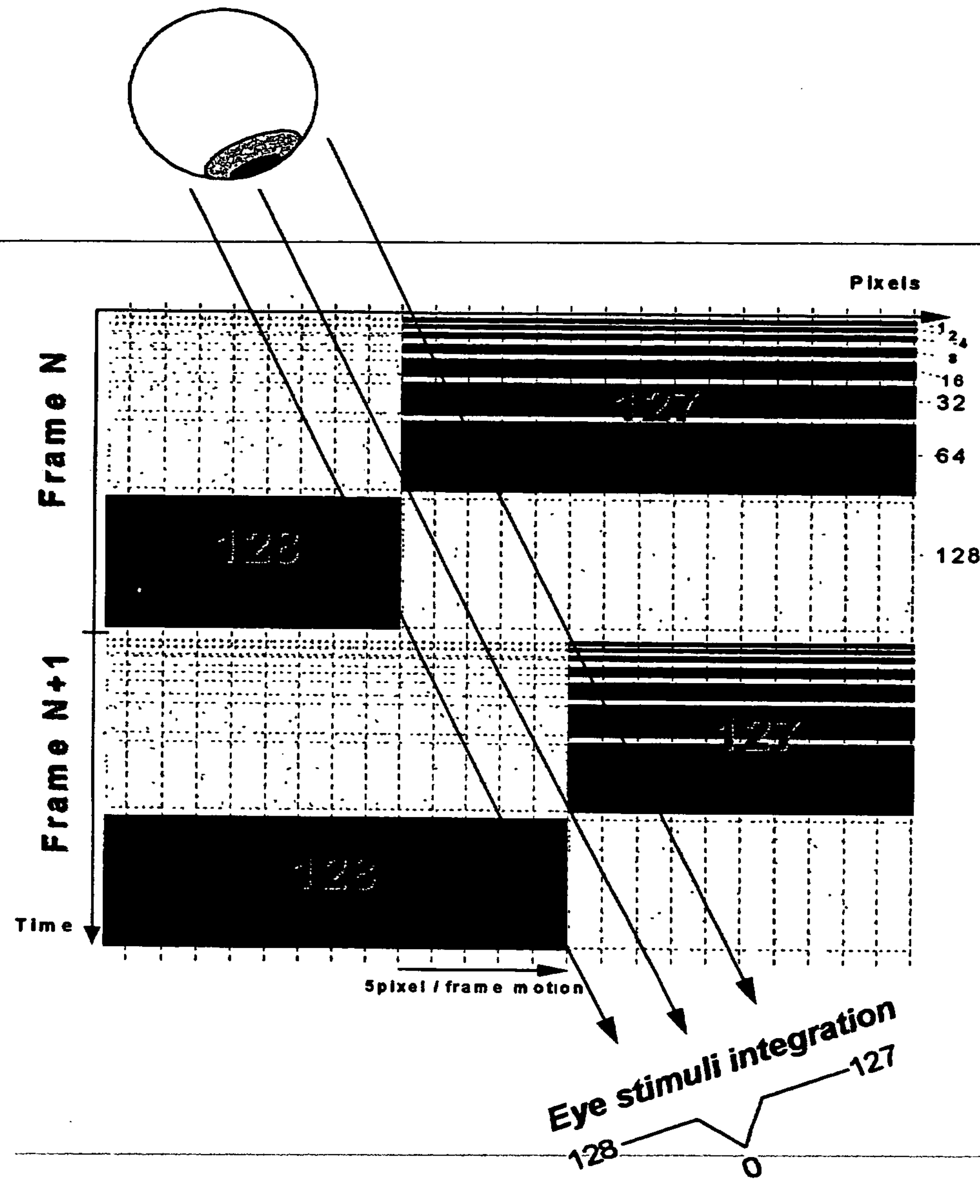


This light emission pattern introduces new categories of image-quality degradation corresponding to disturbances of gray levels and colors. These will be defined as dynamic false contour since the fact that it corresponds to the apparition of colored edges in the picture when an observation point on the PDP screen moves. Such failures on a picture leads to an impression of strong contours appearing on homogeneous area like skin. The degradation is enhanced when the image has a smooth gradation and also when the light-emission period exceeds several milliseconds.

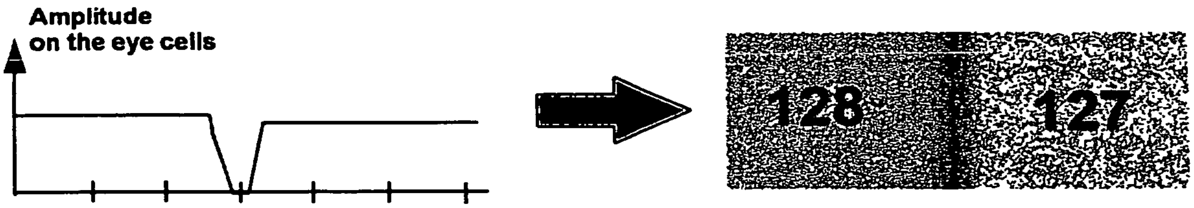
In addition, the same problems occur on static images when observers are moving their heads and that leads to the conclusion that such a failure depends on the human visual perception.

To understand a basic mechanism of visual perception of moving images, a simple case will be considered. Let us assume a transition between the level 128 and 127 moving at 5 pixel per frame, the eye following this movement.

The next figure represents in dark green the lighting sub-fields corresponding to the level 128 and in green, these corresponding to the level 127.



On this last figure, we can follow the behavior of the eye integration during a movement. The two extreme diagonal eye-integration-lines show the limits of the faulty perceived signal. Between them, the eye will perceive a lack of luminance which leads to the appearing of a dark edge like on the next figure :



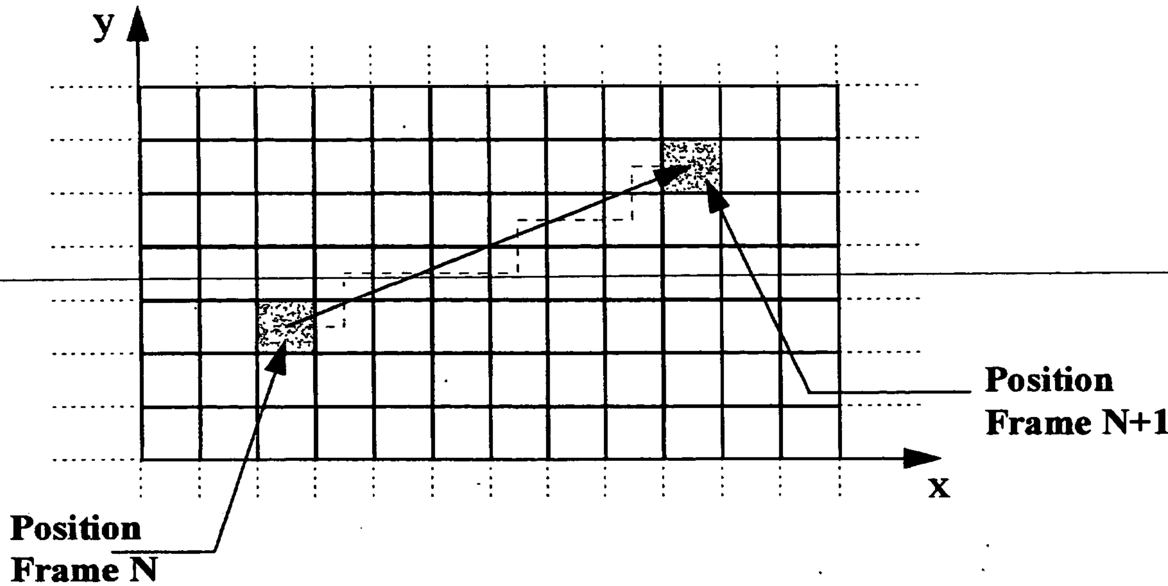
2. Principle of dynamic false contour compensation

The false contour effect is produced on the eye retina when the eye follows a moving object since it integrates right information at wrong places. There are different methods to reduce such an effect but the more serious ones are based on a motion estimator (dynamic methods) which aim to detect the movement of each pixel in a frame in order to anticipate the eye movement or to reduce the failure appearing on the retina through different correction.

In other words, the goal of each dynamic algorithm is to define for each pixel locked by the eye, the way the eye is following the movement during a frame in order to generate a correction on this trajectory.

Consequently, for each pixel of the frame N, we will dispose of a motion vector $\vec{V}=(V_x;V_y)$ which describes the complete motion of the pixel from the frame N to the frame N+1. Nevertheless, the goal of a false contour compensation is to apply a compensation on the complete trajectory. In other words, such an algorithm needs a way to convert this vector in a trajectory on a matrix display.

Taking the example of a vector $\vec{V}=(7;3)$, the next figure shows that the definition of this vector is not enough to determine one trajectory :



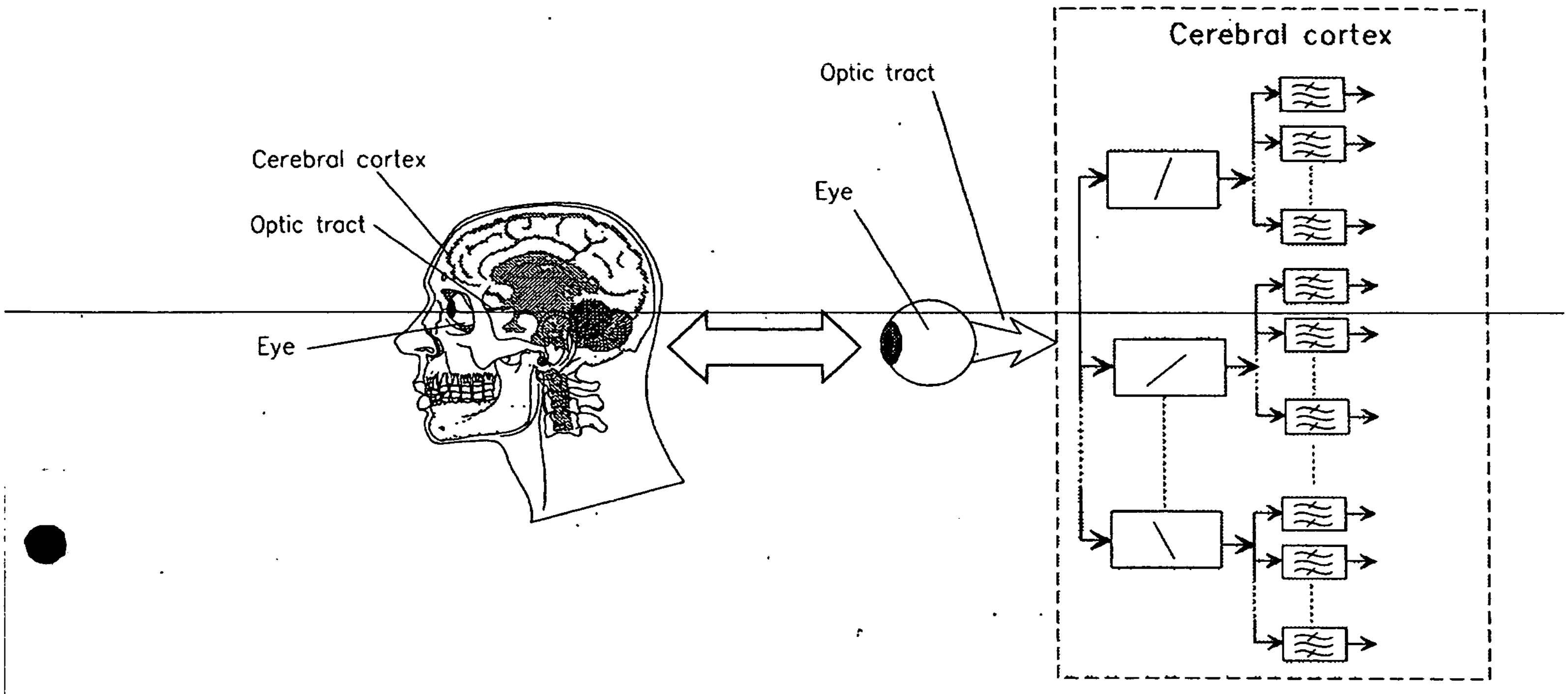
On the previous figure, the vector represents the real motion of a pixel, that means the real trajectory the eye will follow when it locks this pixel. The doted line represents a possible trajectory in the matrix array. Yet, there are different other possible trajectories and it is necessary to define a trajectory as near as possible from the eye integration trajectory. In order to do that, a better understanding of the human visual system is recommendated.

3. Human visual system (HVS): cortex areas

The complete human visual system could be seen as a picture encoder to reduce the informations received by the retina to essential informations which could be rapidly interpreted by the brain.

For instance, the pupil could be seen as a low-pass filter which reduces the amount of high spacial frequencies. The goal of this document is not to make a complete exposition of the human visual system but to extract some important characteristics from the HVS to explain the ideas included in this invention disclosure.

One key point for us is the fact that the cortex areas will analyse the incoming picture with a discrete filter bank as described in the following picture :



This last figure shows that the signal coming from the eye will be analyzed between preferential directions and the number of these direction is limited (discrete analysis). In order to measure the exitation intensity in each direction, each directional filter will be followed through a filter bank in spacial frequencies.

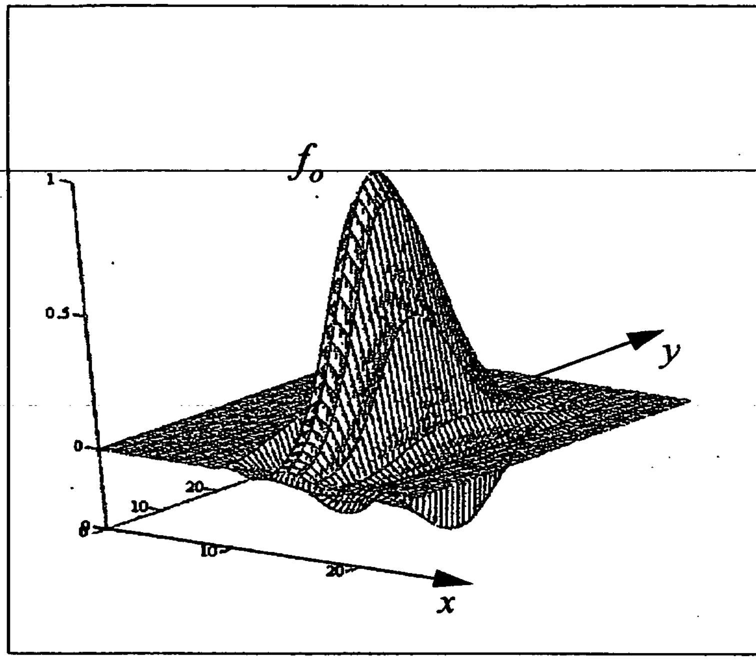
In fact, medical experiments have shown that this decomposition through a filter bank can be seen as a mathematical decomposition in gabor wavelets which are very near from the simple receptor fields of the cortex cells.

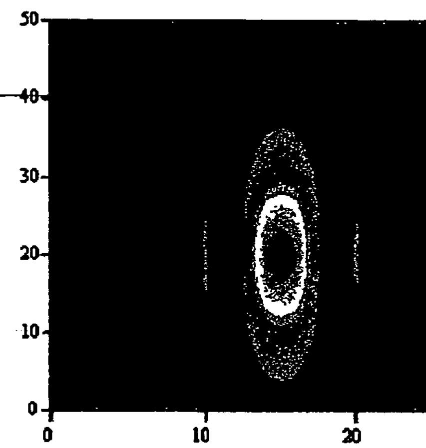
The mathematic formula of such a gabor wavelet is the following one :

$f_o = \exp\left(-\pi\left[(x-x_o)^2 a^2 + (y-y_o)^2 b^2\right]\right) \times \exp\left(-2i\pi\left[u_o(x-x_o) + v_o(y-y_o)\right]\right)$ where (x_o, y_o) represents the position of the filter modulated by the spacial frequencies (u_o, v_o) and with an orientation of $\arctan\left(\frac{v_o}{u_o}\right)$.

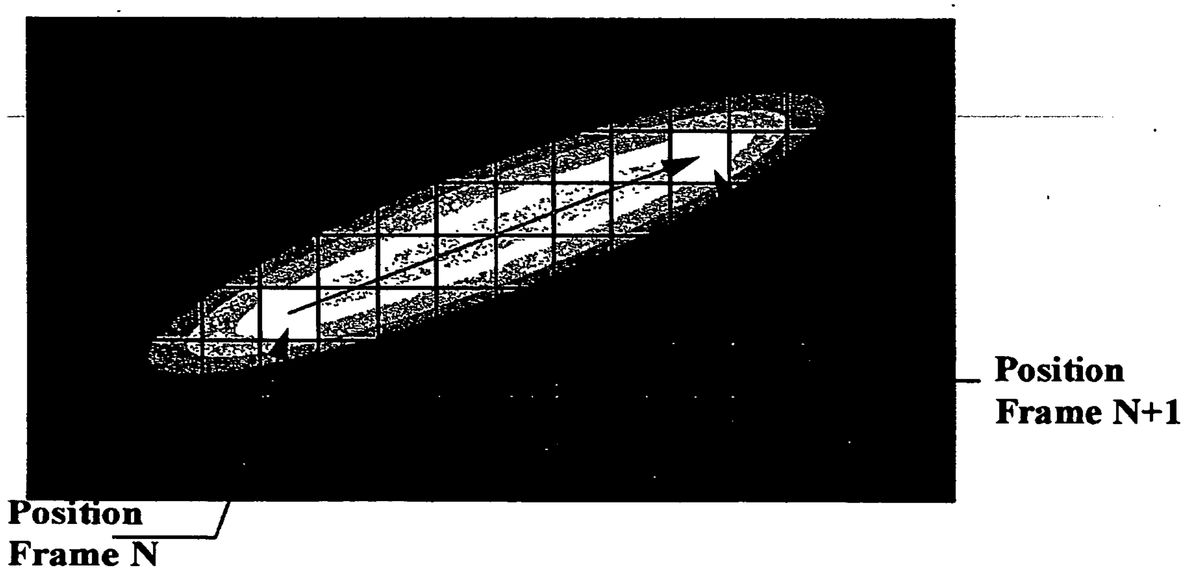
The two following curves represents the aspect of such a gabor function. The first one representing a 3D aspect of the

gabor function, the second one representing a colored 2D aspect of the same function.





These graphics shows how the eye will analyse an object transition or a movement in a special direction.
Let us apply the 2D curve on the previous moving pixel with the vector $\vec{V} = (7;3)$



On this previous picture, the middle area (directly arround the vector) represents the more sensitive one for the eye integration. Consequently, each kind of dynamic false contour compensation should be spread on this area which really define the compensation trajectory.

The basic idea of the algorithm described in this document is to define a vector post-processing method which enables such an adaptated compensation.

4. Algorithm principle

In theory, the different motion vectors could have any kind of values and so any kind of direction.

4.1. The first stage is to convert the two vector components to an integer:

The aim of each compensation is to reduce, in the right direction taking into account the right amplitude of the movement, the false contour effect.

In fact, since the compensation has to be applied on a matrix array of pixels (discrete positions), the two motion vector components have to be integer ones to apply a correction on a discrete trajectory (defined with integers). In that case, it is necessary to round the vector components coming from the motion estimator. Every kind of rounding can be foreseen. Nevertheless, experiments made on different available motion estimator shows that a rounding down could improve the final result. In order to simplify the further explanations a rounding down will be chosen for the rest of this document .

It is obvious that a compensation which is based on a lower value of the movement amplitude but which respects the right direction will still provide a gain in the reduction of the false contour. On the other hand, if the compensated motion amplitude is too high, we will generate a false contour effect in the opposite level to the effect we will try to compensate. In addition, the fact to jump from an under-

compensation to an over-compensation will suddenly change the color of the false contour and that makes it more visible.

Consequently, for all computations one kind of rounding has to be precisely defined and an under-compensation (used motion amplitude lower than the real one) will be chosen for further explanations.

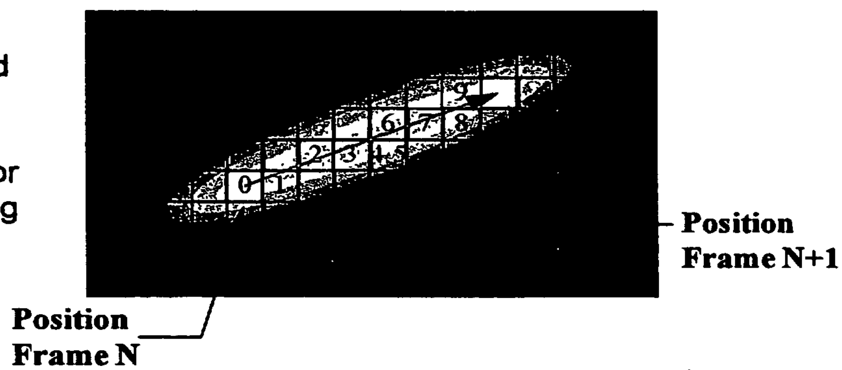
For that reason, the first stage of the algorithm will convert each vector component to an integer with a rounding down to the neariest lower integer:

$$\vec{V}_1 = \vec{V}\left(round \downarrow (V_x), round \downarrow (V_y)\right).$$

4.2. Conversion of standard vectors to basic ones :

Before to go further, an example of a improperly correction will be given to introduce the new computation methodology.

Taking into account a simple correction based on 10 pixels in the case of a trajectory defined with the vector $\vec{V} = (7;3)$ the following correction could be faulty implemented :



On this last figure, the number corresponds to one of the ten elements of the compensation. It is obvious that this compensation does not respect the symmetry of the human eye integration function and leads to a non-optimal false contour correction.

In fact, we have seen in the paragraph 3. that the human cerebral cortex will decompose each movement and stimuli in preferential directions. In fact, since the human visual system does not dispose of an infinite number of such directions, those directions could be defined as discrete ones.

For that purpose one principle of the algorithm is to convert our vectors to a discrete number of direction (convert all vectors to simplest ones which leads to a more symmetrical compensation).

We have seen in the paragraph 4.1. that our vectors are based on integers and consequently, the direction given by each vectors is based on the ratio between the two integer vector components.

In order to build a discrete space of directions, a good possibility is to define each vector as based on two components having an integer ratio inbetween.

For that purpose, the second stage of the processing described here will correspond to a modification of the vector components as described below :

select the smallest vector component: $S = \min(V_x, V_y)$

compute the ratio R between S and the second component:

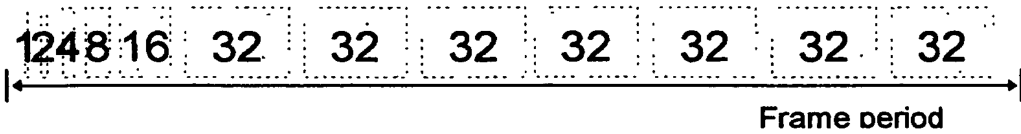
$R = \frac{V_t}{S}$ in which $V_t = \max(V_x, V_y)$

round the ratio R and then update the second vector component: $V_t' = round(R) \times S$

For instance the vector $\vec{V}(7;3)$ will be converted to the vector $\vec{V}'(6;3)$ and the vector $\vec{V}(2;9)$ will be converted to $\vec{V}'(2;8)$. These two vectors are a part of a discrete space of vectors and their form leads to a better symmetry of the compensation.

Now, this processing will be illustrated through an example based on a Sub-field shifting algorithm. This algorithm is explained in detail in the European Patent applications 98114883.6 and 99114587.1 of the applicant. For the disclosure regarding this algorithm it is therefore expressively referred to these patent applications.

Taking into account as example a motion vector defined by $\vec{V} = (7.3;3.7)$ and the case of a 12 Sub-field coding described on the next picture :



Conversion of standard vector components to integers one

$\vec{V}' = (7;3)$

Conversion of the new vector to a basic one

$\vec{V}'' = (6;3)$

Computation of the correction positions

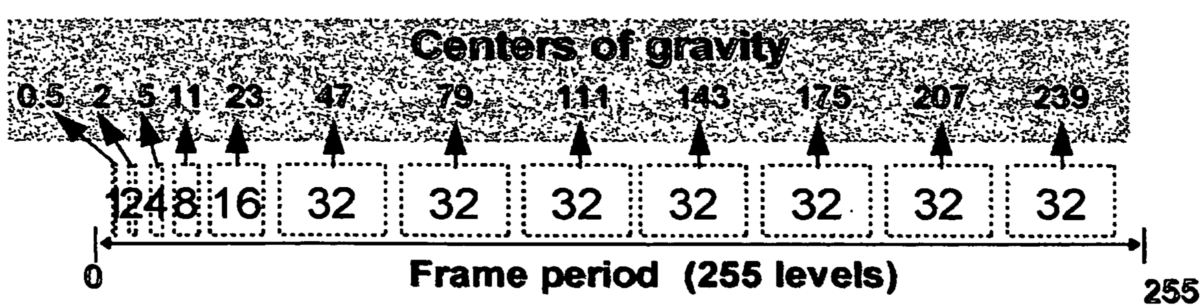
The main idea of the sub-field shifting algorithm is to anticipate the movement in order to position the different bit planes of the moving area on the eye integration trajectory. That means we will shift the different bit-planes depending on the eye movement to make sure the eye receive the right information at the right time. For that purpose we have defined centers of gravity : $G(n) = \sum_{i=1}^{i=n-1} Dur(i) + \frac{Dur(n)}{2}$

in which G() represents the center of gravity location in the frame, n the current sub-field and Dur() the duration of the sub-field. This duration includes the addressing time as following :

$Dur(n) = Tadd + Tn$

in which Tadd represents the duration of the addressing and Tn the duration of the sub-field itself.

For instance, a result of such a computation should be :



Giving a motion vector $\vec{V} = (Vx;Vy)$, the new position of the sub-fields will be so calculated : $\Delta x_n = \frac{Vx \cdot G(n)}{Dur(F)}$ and

$\Delta y_n = \dfrac{Vy \cdot G(n)}{Dur(F)}$ in which, Dur(F) represents the complete duration of the frame.

In our example where $\vec{V} = (6;3)$, we will find the following results :

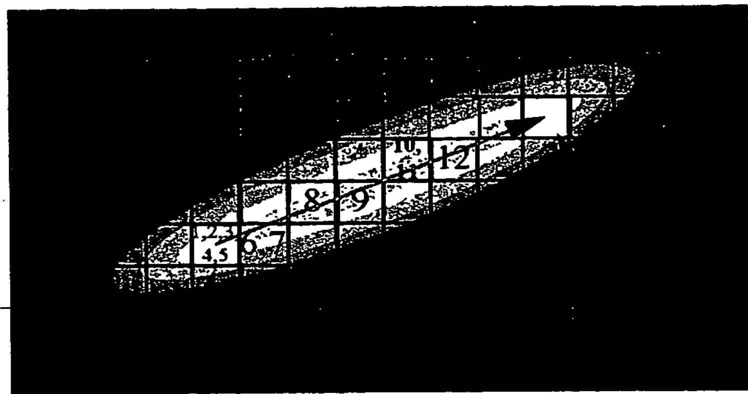
Sub-field N° *i*	1	2	3	4	5	6	7	8	9	10	11	12
Δx^i	0.01	0.05	0.1	0.26	0.54	1.1	1.85	2.6	3.36	4.1	4.87	5.6
Δy^i	0.005	0.02	0.06	0.13	0.27	0.55	0.93	1.3	1.68	2.05	2.43	2.8

Compute how the correction will be applied

Since a correction could only be applied at one single position defined by two coordinates in the matrix of pixel, we will have to round the previous computed values. Different kind of rounding process could be applied and for this example a round down process of each previous computed coordinates has been chosen to respect the choice of paragraph 4.1 to make under-compensation (obviously over kind of compensations are possible) :

Sub-field N° *i*	1	2	3	4	5	6	7	8	9	10	11	12
Δx^i	0	0	0	0	0	1	1	2	3	4	4	5
Δy^i	0	0	0	0	0	0	0	1	1	2	2	2

On the following picture, we can see the result of this compensation:

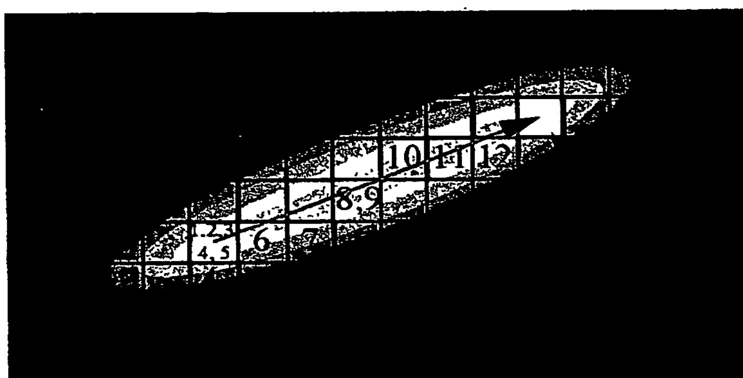


Position Frame N+1

Position Frame N

compared to a standard compensation based on $\vec{V}'=(7;3)$ in which we have implemented only a rounding down process :

Sub-field N° i	1	2	3	4	5	6	7	8	9	10	11	12
Δ_x^i	0.01	0.05	0.14	0.3	0.63	1.29	2.17	3.05	3.92	4.8	5.68	6.56
Δ_y^i	0.006	0.02	0.06	0.13	0.27	0.55	0.93	1.3	1.68	2.05	2.43	2.8



Position Frame N+1

Position Frame N

A comparison of these two last figures shows that the second one does not respect the symmetry of the human visual system. In that case, the experience shows some artefact produced by the compensation itself even in areas where no false contour is visible. In addition, when the vector space is discrete, the compensation stays more stable between two serious vector changes.

4.3. Case of correction made on amplitude basis :

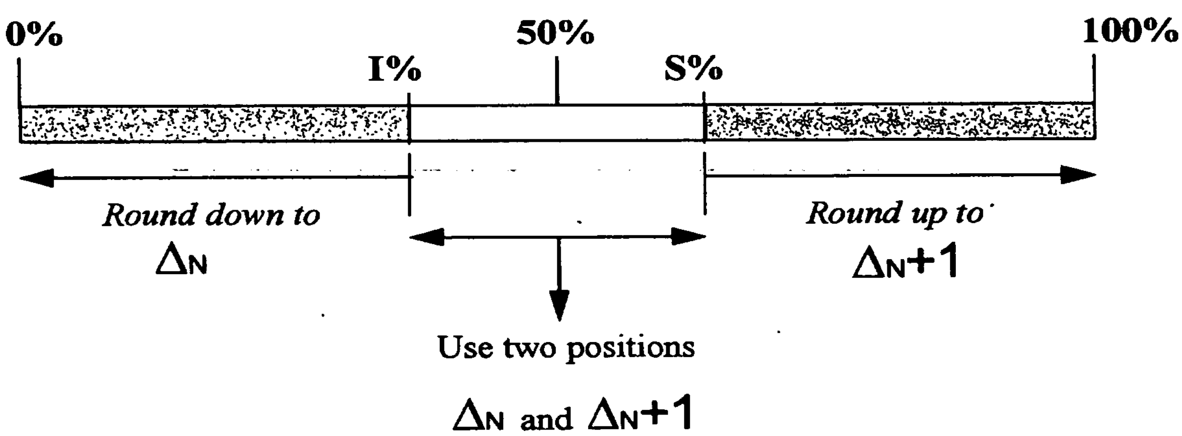
Today a lot of correction based on motion vectors will be applied on amplitude basis. In that case, one possibility is to add a positive or negative signal to the original pictures at different positions depending on the motion vectors (A lot of other kind of corrections could be find in the literature...).

In that case, at each matrix position a combination of two or more corrections can be applied. Consequently, giving a correction, one or two positions for this correction can be defined in the matrix.

Using the assumption of appliing a correction based on N elements to the trajectory defined by the vector $\vec{V}=(V_x;V_y)$.

For instance, a simple way to compute the position $P_i=(\Delta'_x;\Delta'_y)$ is defined by the formula : $\Delta'_x = i\times\frac{V_x}{N}$ and $\Delta'_y = i\times\frac{V_y}{N}$.

Nevertheless, in order to respect the symmetry of the gabor function, a special rounding processing will be applied. This aims to produce an artificial symmetry in the compensation. In that case, each compensation could be applied at one or two positions like described below :



This last figures describe the following processing :

If $\Delta'_\varepsilon = i\times\frac{V_\varepsilon}{N} \le I\%\times round\uparrow\left(i\times\frac{V_\varepsilon}{N}\right)$ then $\Delta'_\varepsilon = round\downarrow\left(i\times\frac{V_\varepsilon}{N}\right)$

If $I\%\times round\uparrow\left(i\times\frac{V_\varepsilon}{N}\right) < \Delta'_\varepsilon = i\times\frac{V_\varepsilon}{N} \le S\%\times round\uparrow\left(i\times\frac{V_\varepsilon}{N}\right)$ then

$$\Delta'_\varepsilon = \left|\begin{array}{l} round\downarrow\left(i\times\frac{V_\varepsilon}{N}\right) \\ round\uparrow\left(i\times\frac{V_\varepsilon}{N}\right) \end{array}\right.$$

If $\Delta'_\varepsilon = i\times\frac{V_\varepsilon}{N} > S\%\times round\uparrow\left(i\times\frac{V_\varepsilon}{N}\right)$ then $\Delta'_\varepsilon = round\uparrow\left(i\times\frac{V_\varepsilon}{N}\right)$

Obviously, the border I% and S% could have different values depending on the compensation algorithm used.

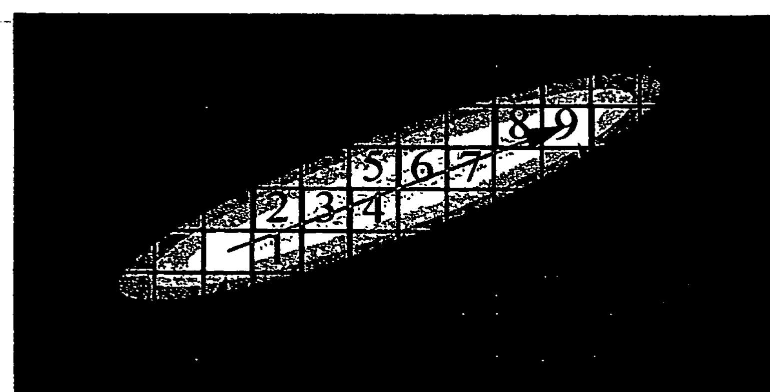
Let us illustrate this processing through a simple example in which the vector $\vec{V}=(6;3)$ and N=9.

Case of I%=40% and S%=60% :

The following table show the computation of the positions for the 9 corrections defined by $P_i=(\Delta'_x;\Delta'_y)$

Correction N° *i*	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Δ'_x	1	1	2	3	3	4	5	5	6
Δ'_y	0	1	1	1	2	2	2	3	3

In that case, the implementation of the correction will look like as on following figure :



Position Frame N+1

Position Frame N

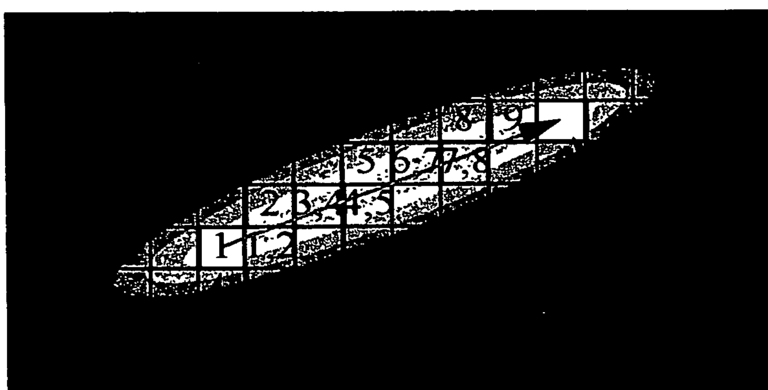
The numbers located in the different pixel blocks refer to the correction signals.

Case of I%=30% and S%=70% :

The following table show the new computation of the positions for the 9 corrections defined by $P_i = (\Delta^i_x;\Delta^i_y)$

Correction N° *i*	1	2	3	4	5	6	7	8	9
Δ_x^i	0, 1	1	2	2, 3	3	4	4, 5	5	6
Δ_y^i	0	0, 1	1	1	1, 2	2	2	2, 3	3

The following figure shows how the compensation will look like :



Position Frame N+1

Position Frame N

Consequently, a change of the values from I and S will have an impact of the density of the correction on the movement trajectory.
In fact, in the cells containing more than one correction, a combination of these corrections will be necessary and will depend of the correction type.

THIS PAGE BLANK (USPTO)

Claims

1. Method for processing video pictures for display on a display device having a plurality of luminous elements corresponding to the pixels of a picture, wherein the time duration of a video frame or video field is divided into a plurality of sub-fields during which the luminous elements can be activated for light emission in small pulses corresponding to a sub-field code word which is used for brightness control, wherein motion vectors are calculated for pixels, **characterized in that,** the motion vector components are rounded to integer values, wherein in the rounding step the components are rounded down irrespective of their rational component value and wherein the motion vectors are used to determine corrected sub-field code words for dynamic false contour effect compensation.

2. Method according to claim 1, wherein the rounded motion vectors are converted to a more symmetrical form before they are used for sub-field code word correction.

3. Method according to claims 2, wherein the following steps are used for motion vector conversion:
   first, the smallest motion vector component S is selected where $S=\min(V_x,V_y)$ with V_x and V_y being the motion vector components;
   second, the ratio R between S and the other motion vector component V_i is calculated, where $R=V_i/S$ and $V_i=\max(V_x,V_y)$, with $i \in [x,y]$;
   third, the ratio R is rounded and the other motion vector component V_i is updated according to the formula $V'_i = round(R)\cdot S$.

4. Method according to one of claims 1 to 3, wherein sub-field shifts are calculated for pixels in dependence of the corresponding motion vectors and wherein a rounding step is performed for each sub-field shift component during sub-field shift calculation, wherein in the rounding step the sub-field shift components are rounded down irrespective of their rational component value.

5. Method according to one of claims 1 to 3, wherein a correction value is calculated for a pixel on pixel level for dynamic false contour effect compensation and wherein the correction value is distributed among a number of pixels which are determined by the calculated motion vector.

6. Method according to claim 5, wherein the pixels $P_i = (\Delta^i_x;\Delta^i_y)$ which are used for correction value distribution are defined by the formulae $\Delta^i_x = i \times \frac{V_x}{N}$ and $\Delta^i_y = i \times \frac{V_y}{N}$, where N is a number according to the number of pixels over which the correction value is to be distributed corresponding to the length of the motion vector $\vec{V} = (V_x;V_y)$, where i is an index running from 1 to N, wherein a specific rounding process is used for correction pixel location, wherein if the rational component value of a vector component Δ^i_x, Δ^i_y lies in a first range, the vector component is rounded down, wherein if the rational component value of a vector component lies in a second range, the vector component is rounded up and down thus leading to two different correction positions in parallel, and wherein if the rational component value of a vector component Δ^i_x, Δ^i_y lies in a third range, the vector component is rounded up.

7